1) Given a set of data points \((x(n), 1 \leq n \leq m)\) we define the convex hull to be the set of all points \(x\) given by
\[
x = \sum_{n=1}^{m} \alpha(n)x(n)
\]
where \(\alpha(n) \geq 0\) and \(\sum_{n=1}^{m} \alpha(n) = 1\). Consider a second set of points \((z(i), 1 \leq i \leq j)\) and its corresponding convex hull. The two sets of points will be linearly separable if there exists a weight vector \(w\) and a scalar \(w_0\) such that \(w^T x(n) + w_0 > 0\) for \(1 \leq n \leq m\) and \(w^T z(i) + w_0 < 0\) for \(1 \leq i \leq j\). Show that, if the two convex hulls intersect, the two sets of points cannot be linearly separable, and conversely that, if they are linearly separable, their convex hulls do not intersect.

2) Let \(x(1) = (1, 1, 1)^T, d(1) = 1, x(2) = (-1, 1, -1)^T, d(2) = -1, x(3) = (3, 0, -3)^T, d(3) = -1, x(4) = (1, -1, 0)^T, d(4) = 1, x(5) = (1, 4, 1)^T, d(5) = 1,\) and \(x(6) = (0, 2, -1)^T, d(6) = -1\) be six binary labeled training examples. These points are linearly separable.

   a) Implement the Perceptron Learning algorithm showing the steps until a solution is reached.
   b) Find the optimum margin classifier (weights and threshold value) and the margin.
   c) Find the least squares SVM solution (weights and threshold value).

3) Prove a corollary to the Perceptron learning algorithm stated in class for nonzero thresholds. For this case show that the number of updates is bounded by
\[
t < \frac{(2R/\gamma)^2}{\nu}
\]
Hint: Proof is similar to zero threshold case, but we need to augment weight vector to include nonzero threshold value.

4) Program the perceptron learning algorithm described in class. Perform the algorithm for \(n = 2, n = 5,\) and \(n = 10\). For each \(n\), consider \(m = n, m = 2n, m = 3n,\) and \(m = 5n\) training examples. For each run, first set solution weight vector \(w^* = (1, 0)^T\) for \(n = 2\) and \(w^* = (1, 0, 0, 0, 0)^T\) for \(n = 5\). Set \(w(0) = 0\) and randomly generate positively labeled examples. Vector components will be drawn from an iid Gaussian random variables with mean 0 and variance 1. Update weight vector as described in class. Once algorithm converges, you will have a weight vector \(\hat{w}\) that will correctly classify \(m\) training examples.
Record the number of updates $k$ and the number of times examples were tested. Generate $10n$ additional random positive examples and test with your $\hat{w}$. Record the percentage that are mislabeled. For each $m$, run the simulation ten times and average all results. From simulating the perceptron learning algorithm what are your observations.

Algorithm for generating 2 iid Gaussian random examples. First generate two independent uniform $[0, 1]$ random variables $X_1$ and $X_2$. Then let $A = \sqrt{-2 \log(1 - X_2)}$. The two iid Gaussian random variables are then $Y_1 = A \cos(2\pi X_1)$ and $Y_2 = A \sin(2\pi X_1)$. If you use matlab, use randn command to generate Gaussian random variables.

5) Consider the data set ps1train. The data set consists of one hundred labeled points with the first two columns representing inputs and the last column representing the output label.

a) Plot the labeled training data. Can a linear hyperplane classify the data correctly?

b) Use a second order polynomial classifier to train the data. From the training data, find the equation describing the classifier. The classifier may not be able to train all data correctly so you will need to have an additional stopping criterion for algorithm after it has gone through so many passes. Find the generalization error by testing the classifier with the data set ps1test.

c) Repeat b) with a third order polynomial classifier.

d) Repeat b) with a fourth order polynomial classifier. What classifier does the best? Discuss your results.